

Vastaako DAD (Disciplined Agile Delivery) perinteisen projektinhallinnan ja ketterän ohjelmistokehityksen yhdistämisessä havaittuihin ongelmiin?

Kirsi Strömsholm

Opinnäytetyö

2014



Tietojenkäsittelyn koulutusohjelma

Tekijä tai tekijät Kirsi Strömsholm	Ryhmätunnus tai aloitusvuosi 2009
Raportin nimi Vastaako DAD perinteisen projektinhallinnan ja ketterän ohjelmistokehityksen yhdistämisessä havaittuihin ongelmiin?	Sivu- ja liitesivumäärä 37 + 2
Opettajat tai ohjaajat Niina Kinnunen	
<p>Opinnäytetyön idea syntyi opinnäytetyön tekijän työharjoittelun aikana. Työharjoittelu- paikassa oli käynnistymässä laaja uuden ERP-ohjelmiston kehitystyöprojekti. Projektia toteutettiin monitiimisesti kolmen organisaation kesken. Organisaatioiden ketteryys, projektiosaaminen ja ohjelmistokehitystaidot vaihtelivat suuresti ja tämä aiheutti projektin myöhästymisen lisäksi ongelmia resursoinnissa ja projektin johtamisessa.</p> <p>Opinnäytetyön tavoitteena on tutkia perinteisen projektinhallinnan ja ketterien menetelmien yhdistämisessä esiintyviä ongelmia ja niiden seurauksia. Toisena tavoitteena on tutkia, vastaako DAD (Disciplined Agile Delivery) viitekehys löydettyihin ongelmiin. Kolmantena tavoitteena on lisätä tietämystä uusimmasta ohjelmistokehityksen suunnasta, jossa yhdistyy perinteisen projektinhallinnan elementtejä sekä ketteryyden menetelmiä ja ideologia.</p> <p>Tutkimus tehtiin kahdessa osassa. Ensimmäiseksi valittiin aineistoksi tutkimuksia, jotka oli tehty organisaatioille, joissa tehtiin ohjelmistokehitystä yhdistämällä perinteinen ja ketterä projektinhallinta. Toisessa osassa tutkittiin, vastaako DAD viitekehys aineistosta löydettyihin ongelmiin.</p> <p>Tutkimuksen perusteella voidaan sanoa, että organisaatioissa on samankaltaisia ongelmia ketteryyden ja perinteisen mallin yhdistämisessä. Soveltava ketteryysosaaminen oli puutteellista aiheuttaen ongelmia resursoinnissa, vaatimusmäärittelyssä, budjetoinnissa ja aikataulutuksessa. Lisäksi asenne havaittiin ongelmaksi keski- ja ylemmän johdon kohdalla. Asenneongelma esiintyi organisaatioissa, joissa komentohierarkia oli ylhäältä alas. Tutkimuksen tulokset osoittivat myös, että DAD:in avulla voidaan vastata yhdistämisessä ilmenneisiin ongelmiin. Avain onnistuneeseen ohjelmistokehitysohjelmaan on ymmärrys ja tietämys käytettävästä elinkaari- tai prosessiviitekehysmallista ja soveltavan osaamisen kasvattamiseksi tarvitaan harjoitusta.</p>	
Asiasanat DAD, Disciplined Agile Delivery, ohjelmistokehitys, ketterä, projektinhallinta	

Degree Programme in Information Technology

Authors Kirsi Strömsholm	Group or year of entry 2009
The title of thesis Does DAD (Disciplined Agile Delivery) respond to problems found combining traditional project management and agile software development?	Number of pages and appendices 37 + 2
Supervisor(s) Niina Kinnunen <p>There was a large ERP software development project starting where the author was doing her internship and the idea to this thesis originated from there. The project was executed by three teams from three companies. There were huge differences between the teams' agile levels, project management and software development know-how. Those differences rendered delay to the project and also problems to resourcing the project and project management.</p> <p>The first objective to this thesis was to study problems combining traditional project management and agile methods and the consequences of those problems. The second objective was to study whether DAD process framework responds to the found problems. And the third objective was to increase knowledge about the newest trends in software development frameworks.</p> <p>The study was made in two parts: firstly on the problems caused when combining traditional project management with agile methods and secondly on finding out whether DAD answers to those found problems.</p> <p>The outcome of the study was that the organizations struggle with the same kind of problems when combining traditional and agile methods. When adapting agility, the problems caused challenges to resourcing, the requirements specification process, budgeting and scheduling. Moreover, problems were found with the attitude of the management in such organizations that are used to operate from an up to down chain of command. The study shows that by implementing the DAD process framework it is possible to solve problems combining traditional project management and agile methods in software development. The key to success in software development based projects is in the understanding and in the knowledge of the used framework. Practice is needed in order to increase applied know-how.</p>	
Key words DAD, Disciplined Agile Framework, software development, agile, project management	

Sisällys

1 Johdanto	1
2 Perinteinen projektinhallinta ohjelmistokehityksessä	4
3 Ketterä ohjelmistokehitys.....	8
4 DAD (Disciplined Agile Delivery)	11
4.1 DAD vaiheet ja päämäärät	15
4.2 DAD vastuut ja rooli	17
4.3 DAD arvojen ja periaatteiden yhteneväisyydet ja erot ketteriin periaatteisiin ..	20
5 Tutkimuksen toteutus, aineisto ja tulokset	24
5.1 Aineiston esittely	24
5.2 Tulokset ketterän ohjelmistokehityksen ja perinteisen projektinhallinnan yhdistämisen ongelmista.....	26
5.3 Vastaako DAD löydettyihin ongelmiin	29
6 Yhteenveto	32
Lähteet	34
Liitteet.....	38
Liite 1	38
Liite 2	39

1 Johdanto

Tämän opinnäytetyön idea sai alkunsa työharjoittelujaksoni aikana havaitsemista ongelmista kehitettäessä uutta ohjelmistoa hajautetusti ja monitiimisesti kolmen eri organisaation voimin. Mitä jos asiakas ei ole projektikypsyyskäs kuin vasta alkutekijöissään ja ketteryydestä ei voi edes puhua vielä, yksi toimittaja on aloittelija ketteryydessään, mutta perinteinen vesiputousmalli luonnistuu ohjelmistokehitysprojekteissa ja toinen toimittaja on pitkän linjan konkari sekä ketteryydessä että projektinhallinnassa?

Ohjelmistokehityksen määrä on kasvanut jatkuvasti teknologian kehittyessä ja perinteiset mallit eivät ole pystyneet vastaamaan muuttuneeseen toimintaympäristöön. Ohjelmistokehitysprojekti poikkeaa huomattavasti esimerkiksi talon rakennusprojektista. Ohjelmistokehityksessä ei välttämättä tiedetä kaikkia sovellusarkkitehtuurin yksityiskohtia suunnitteluvaiheessa. Taloa rakennettaessa asia on toisin, arkkitehtuuripiirroksat ovat tiedossa ja niiden voidaan olettaa olevan muuttumattomia. Projektinhallinta on tietämyksen, taitojen, työkalujen ja tekniikoiden soveltamista, jotta vaatimuksia vastaavaan lopputulokseen päästäisiin (Marchewka 2013, 13). Ohjelmistokehitysprojekteissa projektinhallinta on ollut vuosituhatien alkuun saakka vahvasti perinteisiä projektinhallinnan elinkaarimalleja noudattava ja noin 15 vuoden ajan on enenevässä määrin kehitetty uusia projektinhallinnan malleja.

Ohjelmistokehitysprojektien erityistarpeisiin on 2000 -luvun alusta noussut vastaamaan ketterä ohjelmistokehitys ja projektinhallinta. Ketterä kehitys mielletään sovelluskehityksen viitekehykseksi, mutta projektien onnistumisprosenttien nousun vuoksi (Marchewka 2013,6; The Standish Group 2013, 3) on ketterästä metodologiasta enenevässä määrin alettu hakea apuja myös koko projektin elinkaaren hallintaan yhdistelemällä perinteistä, suunnitelmapainotteista, projektinhallintamallia ja ketteriä käytänteitä. Yksi ajankohtaisimmista prosessiviitekehyksistä on vuonna 2006 alkunsa saanut DAD, Disciplined Agile Delivery, jossa on elementtejä niin ketterästä metodologiasta kuin perinteisestä projektinhallinnasta.

Ei minkään viitekehyksen tai mallin yksityiskohtainen noudattaminen ohjelmistokehityksessä ole sinällään autuaaksi tekevää ja kaikkien ongelmien ratkaisija. Toimintatapo-

jen, käytänteiden ja ideologian sisäistäminen ja käyttöönotto vaatii aikaa ja halua kehittyä. Tätä halua tarvitaan organisaation jokaisella tasolla. Organisaatioiden ylhäältä alas komentohierarkia asettaa ylemmän ja keskijohdon mahdollistajan asemaan. Onnistumisprosentti projektissa, ketterässä, perinteisessä tai yhdistelmässä korreloi johdolta saadun tuen ja kannustuksen määrän kanssa. (The Standish Group 2013, 3).

Vaikka ketteryyden on havaittu nostavan projektien onnistumisprosenttia, on ketteryys myös kääntynyt itseään vastaan. Tämä on havaittu yhteisöissä, joissa ketteryys on nostettu lähes uskonnolliseen asemaan viitekehyksenä, jonka ominaisista käytänteistä ei saa poiketa. (Begel ym. 2013, 9).

Ketteryyden julistuksesta on mennyt yli vuosikymmen ja organisaatioiden omista yhdistelmämalleistakin on jo saatavilla tietoa. Tätä opinnäytetyötä tehdessä oli mielenkiintoista havaita, että perinteisten elinkaarimallien ja ketteryyden yhdistämisessä on organisaatiosta riippumatta samankaltaisia ongelmia. Penttonen (2013) havaitsi tutkimuksessaan, etteivät ketterät metodologiat vastaa koko projektin elinkaaren aikana ilmenneisiin tarpeisiin. Toisaalta perinteiset mallit eivät pysty vastaamaan ohjelmistokehityksen vaatimusten muutoksiin riittävän joustavasti. Tähän tarpeeseen on kehitetty DAD viitekehys ja tämän opinnäytetyön toisessa tutkimusosassa tutkitaan vastaako DAD ongelmiin, jotka ovat ilmenneet perinteistä ja ketterää metodologiaa yhdistettäessä.

Tavoitteina ovat kartoittaa perinteisen projektinhallinnan ja ketterän ohjelmistokehityksen ongelmia, tutkia vastaako DAD löydettyihin ongelmiin, lisätä tietämystä uusimmasta suuntauksesta projektinhallinnassa ja prosessiviitekehyksistä ja niiden käytänteistä ohjelmistokehityksessä. Tässä kohtaa tietämyksellä tarkoitetaan Markkulan, Mustajärven ja Suurlan (2002, 35) määritelmää tietämyksestä; tietämyksen perustana on tieto, joka on prosessoitu yksilön henkilökohtaiseksi ja tiedolliseksi rakenteeksi, informaatioksi. Kun informaatio on osa kontekstia, voidaan puhua tietämyksestä. (Markkula ym. 2002, 35.) Edellä mainittua tietämystä tarvitaan, jotta ketterän ja perinteisen metodologian yhdistämisellä tai soveltamisella saavutetaan suurempi onnistumisprosentti ohjelmistokehitysprojekteissa. Tietämystä tarvitaan myös siinä, kuinka yhdistäminen ja soveltaminen on kannattavinta tehdä. Tosin sanoen voidaan ajatella, että kahden metodologian soveltaminen yhdistelmämalliksi tai hybridimallin soveltaminen tarkoittaa meto-

dologioiden syvempää hallintaa. Tavoitteeseen pyritään etsimällä vastauksia kirjallisuudesta, opinnäytetöistä, internetistä ja yliopistotutkimuksista.

Tässä opinnäytetyössä ei paneuduta mihinkään yksittäiseen perinteiseen tai ketterään malliin tai viitekehykseen, koska näistä on saatavilla tietoja ja tutkimuksia pilvin pimein. Lukijalla odotetaan olevan perustietämys ohjelmistokehityksestä, projekti- ja systeemi-työstä, ketteryydestä ja niihin liittyvistä termeistä. Sen sijaan ketterän sukupolven seuraajan kurinalaisen ketteryyden viitekehys, DAD (Disciplined Agile Delivery), avataan yksityiskohtaisemmin, koska asia on verrattain uusi.

Opinnäytetyössä pyritään vastaamaan kirjallisuuden, internet -lähteiden, kuten blogit ja aikaisempien tutkimusten avulla seuraaviin tutkimuskysymyksiin:

- Mitkä ovat perinteisen projektinhallinnan ja ketterän kehityksen yhdistämisen kriittiset ongelmat ohjelmistokehityksessä?
- Mitä ongelmat aiheuttavat?
- Mikä on DAD ja kuinka se eroaa ketteristä?
- Vastaako DAD perinteisen projektinhallinnan ja ketterän sovelluskehityksen yhdistämisessä havaittuihin ongelmiin?

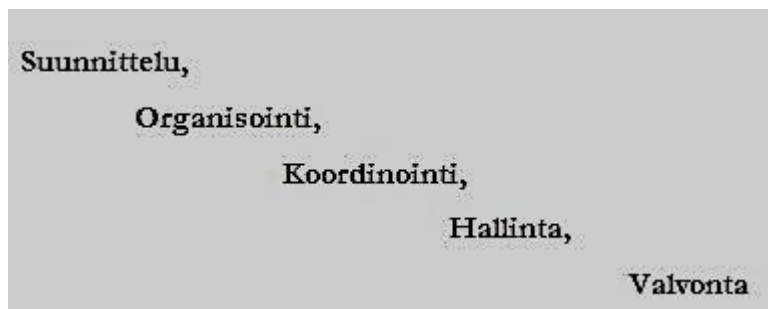
Tämä opinnäytetyö sisältää kaksiosaisen tutkimuksen. Ensimmäisessä osassa tutkitaan ketteryyden projektinhallinnallisia ominaisuuksia ja perinteisen projektinhallinnan ja ketterän kehityksen yhdistämistä ja niiden mahdollisia ongelmia. Toisessa osassa tutkitaan vastaako DAD ensimmäisessä osassa ilmenneisiin ongelmiin. Näkökulmana on ohjelmistokehitysprojekti. Ohjelmistotyyppiä ei ole määritelty tarkemmin.

2 Perinteinen projektinhallinta ohjelmistokehityksessä

Projektit alkavat tarpeesta kehittää tai muuttaa jotakin. Ensiksi määritellään tavoite ja analysoidaan edellytykset. Sen jälkeen valitaan ratkaisu ja työtapa. Tämän jälkeen työ suunnitellaan ja organisoidaan. Toteutusvaihe sisältää seurannan, tarvittaessa tarkennukset ja lopuksi toimituksen ja toivottavasti käytön. Lopuksi projekti lopetetaan. (Tonnquist 2012, 31.)

Tonnquist (2012, 80) esittää projektissa olevan kolme parametria, jotka edustavat aikaa, laatua ja resursseja. Näistä vain yksi voi olla määräävä. Toisin sanoen, jos aika on tärkein parametri, on aikaparametrin arvo lopetuspäivä kaikkein tärkein huomioitava tekijä. Tulos- ja kustannusparametrit mukautetaan tärkeimmäksi valittuun parametriin.

Perinteinen projektinhallinta noudattelee yleisesti hyväksyttyjä Henry Faoylin 1916 julkaistuja johtamisperiaatteita. Periaatteissa esitetään asiat, joita tarvitaan tehokkaaseen johtamiseen. Nämä viisi elementtiä ovat esitelty alapuolella olevassa kuviossa 1. (Forsberg, Mooz & Cotterman 2000, 19.)



Kuvio 1. Henry Faoylin viisi johtamiselementtiä

Forsberg ja muut (2000, 19-20) esittelevät Faoylin 14 periaatetta yhdistettynä viiteen elementtiin tarjoavan struktuurin projektin ohjaamiselle:

1. Työn jakaminen
2. Vastuun ja vallan tasapaino
3. Kuri

4. Ohjauksen yhtenäisyys
5. Suunnan yhtenäisyys.
6. Henkilökohtaisten tavoitteiden alistaminen yleisille tavoitteille
7. Vastike henkilöstöstä
8. Keskittäminen
9. Yksiselitteinen ketju (määräysvallan)
10. Järjestys
11. Oikeudenmukaisuus.
12. Henkilöstön vakaa työpaikka
13. Aloitekyky
14. Ryhmähenki.

Projektinhallintamallit tai -rakenteet varioivat monin eri tavoin. Suosittuja käytössä olevia esityksiä projektin elinkaariksi ovat ympyrä-, vesiputous-, spiraalimallia ja USAn liittovaltion puolustusministeriön (DoD¹) käyttämä rakenne, joka kuvaa ohjelmistoon liittyviä tapahtumia erillisenä laitteiston tapahtumista. (Forsberg ym. 2000, 21.)

Marchewka (2013, 35) esittää järjestelmäkehitykselle elinkaarimallia, jossa yhdistyvät projektinhallinnan viisi vaihetta ja toteutusvaiheen järjestelmäkehityksen elinkaarimalli. Projektinhallintakokonaisuus koostuu seuraavista vaiheista: 1. määrittele projektin päämäärä, 2. suunnittele projekti, 3. toteuta projekti, 4. sulje projekti ja 5. arvioi projekti. Järjestelmäkehitys sisältyy toteutusvaiheeseen, joka sisältää kartoituksen, analysoinnin, suunnittelun, toteutuksen ja ylläpito- ja tukivaihe.

Clarce, O'Connor & Yilmaz (2012, 10-11) havaitsivat tutkimuksessaan, että tarkasteltaessa rooleja ja vastuita perinteisessä ja ketterässä ohjelmistokehityksessä, ovat samat vastuut havaittavissa molemmissa malleissa. Erot voidaan havaita roolinimikkeissä liitteessä 1.

¹ Department of Defense

Perinteiset projektinhallinnan elinkaarimallit tunnetaan jo hyvin. Perinteinen ei tarkoita kurinalaista, mutta sen sijaan se voi tarkoittaa liiallista byrokratiaa ja liian yksityiskoh- taista mallin seuraamista (Ambler & Lines 2012, 2). Forsberg ja muut (2000, 20) katso- vat, että Faoylin 14 periaatteesta neljän huomiotta jättäminen aiheuttavat yleisellä tasol- la suosittujen elinkaarimallien ongelmat. Syiksi pois jättämiselle on sanottu, että jotkut ajattelevat niitä ei-toivottavina tai arvioivat niiden olevan mahdotonta saavuttaa projek- tiorganisaatiossa. Nämä neljä ovat vastuun ja vallan tasapaino, ohjauksen yhtenäisyys, yksiselitteinen määräysketju ja pysyvyys.

Perinteisissä ja suosituissa elinkaarimalleissa on havaittavissa teknisten seikkojen hallin- nan heikkous. Esimerkiksi saatetaan erotella laitteistoon liittyvät tapahtumat ohjelmis- toon liittyvistä tapahtumista, aloittaa ohjelmiston koodaus vasta varsin myöhäisessä vaiheessa elinkaarta tai elinkaarimalli tarjoaa epäselvän käytänteen kehittämisen hallin- nan tarkastuksiin. (Forsberg ym. 2000, 20-24.)

Suosituissa kaksiulotteisissa elinkaarimalleissa nähdään heikkoutena myös tapahtumien tarkastelu peräkkäisinä tapahtumina, esimerkiksi riskianalyysi ja riskienhallinta. Projek- tien tapahtumien tarkastelu vain toinen toistaan seuraavina tapahtumina vääristää ja yksipuolistaa kuvaa projektinhallinnasta prosessina. (Forsberg ja muut 2000, 20.)

Jos projekti ei omista resurssejaan tai projektityöntekijä suorittaa projektiroolinsa mu- kaisten tehtävien lisäksi linjaorganisaation työtehtäviä, projektien ongelmaksi saattaa usein muodostua projektin resurssien jakaminen toisten projektien tai päivittäisen liike- toiminnan kesken. Tämä saatetaan huomata projektin edistymisen hidastumana. (Tonnquist 2013, 21; Laanti 2013, 51.)

Etupainotteinen suunnittelu yhdessä tiukan vaiheistuksen kanssa aiheuttavat monien projektien epäonnistumisen, koska asiakas ei kykene antamaan vaatimuksia kerralla aivan alkuvaiheessa (Kuikka 2008, 73). Lisäksi, jos projektin alustusprosessi jätetään heitteille, muodostuu ongelmaksi se että projektit aloitetaan ilman selviä tavoitteita ja kuva päämäärästä on epäselvä (Tonnquist 2013, 40).

Perinteisen projektinhallintamallin ongelmaksi pienemmissä organisaatioissa muodostuu sen massiivisuus ja raskaus soveltaa käytäntöön. Pienemmissä yrityksissä yhdellä henkilöllä on monesti useita eri rooleja ja näin ollen perinteisen mallin vastuut saattavat kasautua yhdelle ihmiselle. (Haarala & Mäkilouko 2013, 86.)

Projektihallinta on perinteisesti nähty olevan seuraavien avainroolien vastuulla: projektin omistaja, projektin ohjausryhmä, prosessin omistaja, projektipäällikkö ja projektitoimisto (ICT Standard Forum 2012). Edellä mainittujen vastuulla on projektin liiketoiminta, projektisuunnitelman hyväksyntä (projektin omistaja), luoda onnistumisen mahdollisuudet projektin onnistumiselle (ohjausryhmä), projektisuunnitelma, operatiivinen johtaminen ja projektista raportointi (projektipäällikkö) ja johtaa valitun projektinhallintamallin mukaista prosessia (prosessin omistaja) (Suomen Projekti-Instituutti 2013). Perinteisessä ohjelmistokehityksessä on johdon vastuulla ohjelmistokomponenttien, jotka ovat jaettu tiimeittäin, välisistä rajapinnoista huolehtiminen (Auer ja muut 2013, 10-11).

3 Ketterä ohjelmistokehitys

Ketteryyden on sanottu olevan uusi ajattelutapa, jonka kautta tiimi tuntee itsensä voimaantumemmaksi, sitoutuneemmaksi ja halukkaaksi parantamaan liiketoimintakykyä (Hartnett, 2011). Lisäksi ketterä liike² tarjoaa vaihtoehtoja perinteiseen projektinhallintaan ohjelmistokehityksessä. Tyypillisesti ketterää lähestymistapaa käytetään, jotta liiketoiminta voisi paremmin vastata ennakoimattomuuteen ohjelmistokehityksessä. (Szalvay & James 2013.) Projekti voidaan dokumentoida ja toteuttaa asteittain. Vaatimuksia tarkennetaan sitä mukaan kun tietämys ja kokemus kerääntyvät. (Tonnquist 2012, 91.)

Ketteryys tarkoittaa läheistä yhteistyötä kehittäjien ja sidosryhmien välillä, jatkuvaa ja säännöllistä ohjelmistoversiota, joka lisää arvoa organisaatiolle, laatuun keskittymistä, arvoa tuovien asioiden tekemistä ja arvoa tuottamattomien asioiden tekemättä jättämistä sekä pyrkimystä parantaa omaa toimintaa koko elinkaaren ajan (Ambler & Lines 2012, 1).

Ketterässä kehityksessä suositellaan mm. että tiimien välillä keskusteltaisiin suoraan esimerkiksi moduleiden välisistä riippuvuuksista. Tämä säästää aikaa ja vähentää väärinymmärryksiä, koska tekijät oletettavammin tuntevat toteutuksen yksityiskohdat tarkemmin. (Auer ym. 2013, 11.)

Ketterissä menetelmissä asiakkaan panos ja osuus ovat merkittäviä. Asiakkaan roolilla on monia eri nimikkeitä, mutta nimikkeestä riippumatta asiakkaan aktiivinen osallistuminen onnistuneen projektin yksi kulmakivistä. (Penttonen 2013, 91.)

Ketterien menetelmien käyttö ei ole täysin mutkatonta tai helppoa. Ongelmiksi on nimetty kokonaisuuden hallinta, tehtävien osittaminen, työmäärän arvioiminen niin

² Suomeksi sanoista: The Agile movement

sprintti- kuin tehtäväkohtaisesti, roolien vastuut ja tehtävät, menetelmän mukainen dokumentaatio, backlogin priorisointi ja tapaamisten kestot. (Roukala 2011, 53.)

Penttonen (2013, 91) kirjasi tutkielmassaan käytännön ongelman tutkimissaan ketterissä metodologioissaan. Tämä kriittinen ongelma oli asiakkaan osallistuminen riittämättömällä tasolla.

Lehtonen (2013, 38) on todennut tutkimuksessaan, että ketteriä menetelmiä sovellettaessa nousi esiin osaamisen rajallisuus ja soveltamisen vaikeus. Dokumentaation tekeminen on havaittu ongelmalliseksi, käytännössä käy helposti niin, että käyttötapaukset kirjataan valmiiksi hyvin aikaisessa vaiheessa ja että liiketoiminnan asiantuntijat kuitenkin tekevät yksityiskohtaista määrittästyötä samoin kuin tehdään suunnitelmapainotteissa elinkaarimalleissa.

Ketterien menetelmien haasteeksi on esitetty myös vaatimusmäärittely. Tästä seuraa kokonaiskustannusten ja keston arviointiongelmia, riittämätön tai vääränlainen arkkitehtuuri, ei-toiminnallisten toiminnallisuuksien riittämätön huomiointi, asiakkaan riittämätöntä mukanaoloa, yksiulotteista vaatimusten priorisointia, vaatimusten vajavaista verifiointia ja liian vähäistä dokumentaatiota. (Ruuska 2012, 68.) Myös Sethi (2013) toteaa blogikirjoituksessaan, että erityisesti ketterissä menetelmissä on vakavia ongelmia ei-toiminnallisten vaatimusten määrittelyssä, koska suurimmassa osassa ketterien menetelmien prosesseja ei tunnusteta ei-toiminnallisten vaatimusten monimutkaisuutta todellisessa tuotantoympäristössä.

Riskien hallinta on puutteellinen monissa ketterissä metodologioissa. Riskien huomiointi keskittyy usein vain työstettävään iteraatioon. Projektin kattava riskianalyysi jää puuttumaan monista ketteristä menetelmistä. (Penttonen 2013, 90.)

Tonnquist (2012, 82) varoittaa, että ketterän menetelmän valinta korvaamaan koko projektinhallinta saattaa olla virhe, joka aiheuttaa ongelmia kokonaisuuden hallinnassa ja näin aiheuttaa aikataulun venymisen ja budjetin ylittymisen.

Liitteenä 1 olevasta kuviosta voidaan havaita, että vastuut ja osaamiset ovat lähestulkoon samat, oli kyseessä perinteinen ohjelmistokehitys tai ketterä. Ketterä suuntaus on omaksunut vastuuta eri roolinimikkeillä kauemmin käytössä olleilta, perinteisiltä malleilta. Keskeisiä ketteriä rooleja ovat mm. scrum master, coach, tester, product owner, customer, programmer ja management. (Clarce ym. 2012, 12.)

Ketteriä menetelmiä voidaan kuvata sanoilla empiirinen prosessimalli. Tällä tarkoitetaan kokemuksiin perustuvaa ja yleensä toistavaa-lisäävää työtapaa. Ketterässä lähestymistavassa projektin johtaminen painottuu yhteistyöhön ja ihmisiin. Tiimissä on yhteistyökykyisiä moniosaajia. Tiimien vastuulla mm. ohjelmointi, suunnittelu. (Lekman 2013, 40; Ruuska 2012, 67.)

Ketterien menetelmien yksi vastuutehtävä on käydä jatkuvaa jacobinpainia³ toteuttaanko tämä vai tuo tai viilataanko vielä tätä vai siirtytäänkö seuraavaan kysymysten välillä (Laanti 2013, 50). Usein tämän vastuun täyttää tuotteen omistaja.

Ketteriä menetelmiä käytettäessä asiakkaan vastuu ja rooli korostuu. Asiakkaan proaktiivinen mukana olo jatkuu koko projektin elinkaaren ajan. Asiakkaalla on monia roolinimikkeitä. Vastuut ovat pääsääntöisesti liiketoimintapäätösten tekeminen. (Clarce ym. 2012, 7, 8; Penttonen 2013, 91.)

Ketteryyden yksi vastuu on kommunikaatio. Se on jokaisen roolin vastuulla, eikä sitä voi korvata millään työkalulla. Ketteryyden tavoite on tuottaa asiakkaalle tuote paremmin, nopeammin ja vähemmän resurssein, jotta päästään tähän tavoitteeseen vaati se kommunikaatiokulttuurin muutoksen pois yksioikoisesta käsky-vastuu-raportointi kulttuurista. (Auer ja muut 2013, 12.)

³ Jaakobin paini vertaus pohjaa raamatun tarinaan, jossa Jaakob painii koko yön tuntemattoman miehen kanssa ja kuvainnollisesti se tarkoittaa suurta sisäistä taistelua ajattelemalla.

4 DAD (Disciplined Agile Delivery)

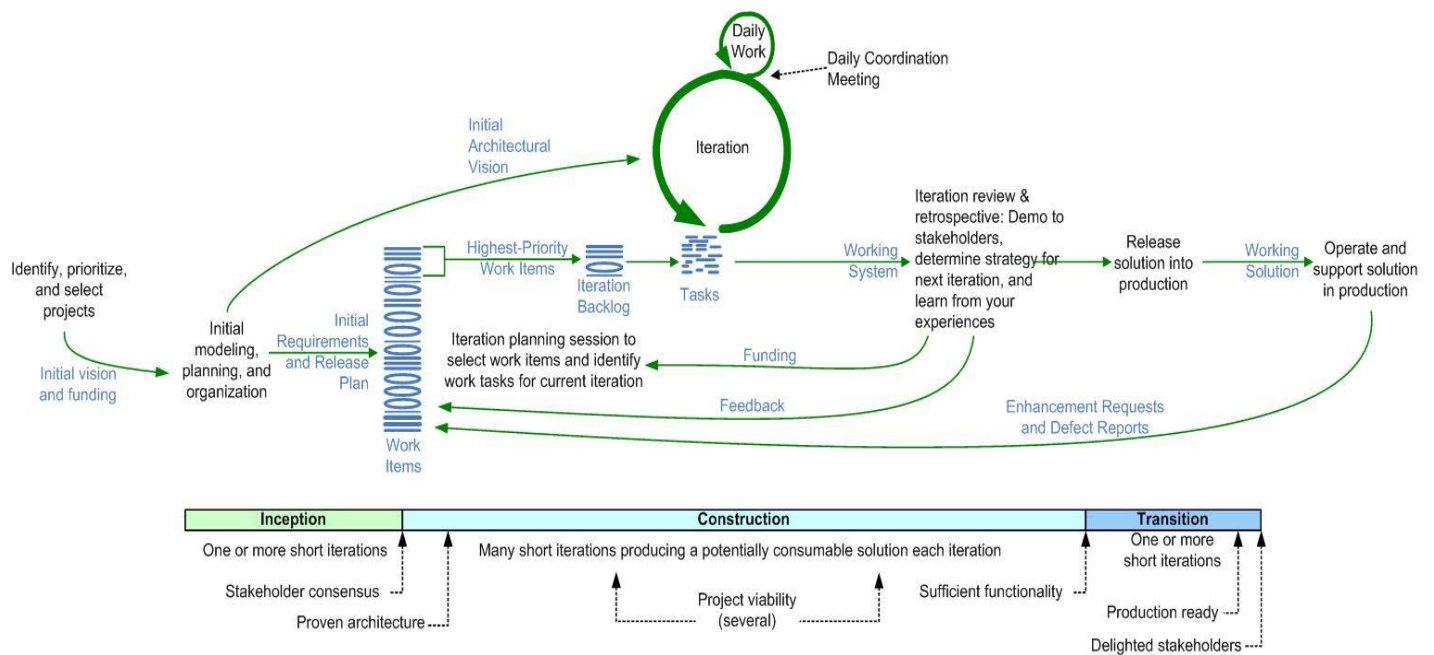
DAD (Disciplined Agile Delivery) käsite on saanut alkunsa vuonna 2007 Scott W. Amblerin työskennellessä johtavana valmentajana maailmanlaajuisesti IBM:n Agilen ja Leanin metodologian ymmärtämisessä ja käyttämisessä. Hän huomasi uudestaan ja uudestaan organisaatioilla olevan vaikeuksia omaksua tunnetuimpia ketteriä käytänteitä kuten Scrum ja XP. Samaan aikaan Mark Lines työskenteli tahollaan samanlaisissa tehtävissä ja huomasi saman vaikeuden; organisaatioilla oli vaikeuksia sopeuttaa ketterien menetelmien kaoottista järjestystä (chaordic⁴) käsky-kontrolli kulttuuriinsa. (Ambler & Lines 2012, johdanto).

Ketterillä menetelmillä saadaan parempia tuloksia verrattuna perinteisiin malleihin ja lähitulevaisuudessa useat yritykset ovat ainakin ajatelleet siirtyvänsä niihin. Valitettavasti valtavirta agilemenetelmät vastaavat vain osaan ohjelmistokehitysprosessin koko prosessin elinkaaren tarpeista. DAD on hybridimalli, jossa yhdistyvät ketterien menetelmien ja leanin menetelmän yleiset käytännöt sekä perinteisien menetelmien merkkipäilykäytännöt. (Ambler & Lines 2012, 23).

Perinteiset projektinhallintamallit sekoitetaan olevan kurinalaisia. Totuus on, että usein niissä rehoittaa käsistä riistäytynyt byrokratia. Toisaalta ketteriä menetelmiä käytetään väärin käyttämällä menetelmää vähän tai ei ollenkaan. (Ambler & Lines 2012, 2.)

Disciplined Agile Delivery (DAD) menetelmän viitekehys, joka tarkoittaa käsitteitä päämääräsuuntautunut (goal-driven), ihmiset ensimmäiseksi (people first) oppimissuuntautunut (learning oriented), ketteriä (agile), hybridi (hybrid), IT -ratkaisu keskittynyt (IT solution focused), toimituskeskittynyt (delivery focused), yritystietoinen (enterprise aware), riski ja arvo suuntautunut (risk and value driven) ja skaalautuva (scalable) (Ambler & Lines 2012, 5).

⁴ Suomennettu kaoottinen järjestys. Chaotic+order = chaordic.



Kuvio 2. DAD elinkaari (Ambler 2010)

Ensimmäisen vaiheen päämääriin lukeutuvat alustavan tiimin muodostaminen, projektin vision määrittely, sidosryhmien konsensus projektin visiosta, yrityksen linjan noudattaminen, alustavan teknisen toimintasuunnitelman, alustavien vaatimusten ja alustavan julkaisusuunnitelman tunnistaminen, työskentely-ympäristön pystyttäminen, rahoituksen turvaaminen ja riskien tunnistaminen. (Ambler & Lines 2012, 13.)

Toisen vaiheen tavoitteet ovat mahdollisesti käytettävän ratkaisun tuottaminen, sidosryhmien tarpeiden muuttuminen käsittely, siirtyminen kohti käyttöönotettavampaa versiota, laatutason ylläpito tai parantaminen ja todenna arkkitehtuuri varhain (Ambler & Lines 2012, 11).

Kolmannen vaiheen päämäärät ovat varmistaa ratkaisun olevan tuotantoon meno valmis, varmistaa sidosryhmien valmius vastaanottamaan ratkaisu, ratkaisun tuotantoon käyttöönotto (Ambler & Lines 2012, 11).

Ihmiset ensin sanaparilla viitataan ihmisten toiminnan korostamiseen mahdollistajana. Perinteinen asiakirjojen toimittaminen työn tuloksista ei ole paras tapa välittää tietoa. Vaarana pullonkaulat ja todistetusti tällä tavalla hukataan aikaa ja rahaa. Lisäksi dokumenttien välityksellä jaettu tieto on helposti väärinymmärretty. Ketterässä toimintata-

vassa minimoidaan asiakirjojen välityksellä tiedon jakaminen ja lisätään tiimin kesken tapahtuvaa toimintaa. Se miten ihmiset työskentelevät ja tekevät yhteistyötä on määräävin tekijä määriteltäessä IT ratkaisun toimittamisen onnistumista. Tämä ajatus sisältyy DAD:iin. (Ambler & Lines 2012, 5-6.)

DAD tiimin jäsenen tulisi omata itsekuria ja tehdä työtä, jonka pystyy saamaan valmiiksi ja tehdä se työ mahdollisimman tehokkaasti, järjestelmällisyyttä, jonka avulla arvioivat ja suunnittelevat työnsä ja tekevät sen iteratiivisesti ja yhteistyössä, itsetietoisuutta, jonka avulla tunnistavat, mikä on itselle paras tapa toimia ja oppivat siitä ja sopeuttavat omaa toimintaansa oppimansa mukaisesti (Ambler & Lines 2012, 7).

Ambler ja Lines (2013, 7) ovat huomanneet, että tehokkaimpia ovat organisaatiot, jotka panostavat henkilökuntansa oppimista edistävään ympäristöön eli ovat **oppimissuuntautuneita**. Oppimista tulee käsitellä kolmesta näkökulmasta. Nämä näkökulmat ovat toimialaoppiminen⁵, prosessioppiminen ja tekninen oppiminen. DAD sisältää monia käytäntöjä oppimiseen, esim. toimialaoppimiseen tarjotaan alustavien vaatimusten visiointi ja inkrementaalinen toimitus potentiaalisesti käytettävästä ratkaisusta, prosessioppimiseen tarjotaan laajaa retrospektiä, jossa tiimi tunnistaa parannuskohteet ja seuraa parannusten onnistumista ja tekniikkaoppimiseen tarjotaan mahdollisuutta etsiä uusia tai parempia tekniikoita. Nämä edellä mainitut käytänteet ovat tuttuja ketteristä menetelmistä. DAD vie oppimisen hieman syvemmälle yritystietoisuuden kautta. DAD selvästi pyrkii parantamaan ja edistämään organisaation ekosysteemiä. DAD tiimien tulee käyttää hyödykseen muiden tiimien ”lessons learned” -kokemuksia ja käyttää aikaa omien kokemusten jakamiseen.

DAD viitekehys pitää kiinni Agile Manifeston periaatteista ja arvoista, koska iteratiivisesti tai **ketterä**sti toimivat tiimit ovat tuottaneet korkeampi laatuista ratkaisuja, parempaa sijoitetun pääoman tuottoa ja tyytyväisempiä sidosryhmiä verrattuna perinteistä mallia noudattaviin tai ad - hoc tiimeihin. (Ambler & Lines 2012, 8.)

⁵ Domain learning

DAD on **hybridi** viitekehys, johon on otettu toimintatapoja useista eri viitekehyksistä. DAD:in käyttö ei perustu kaikkien käytänteiden käyttöön per projekti vaan jokaiselle projektille valitaan omat käytännöt, käytännöt, jotka parhaiten sopivat kyseessä olevalle organisaatiolle. DAD sisältää rutineja ketteristä viitekehyksistä kuten Scrum, XP, Agile Modeling, Agile Data, IMB practices, Open Unified Process, AUP, Crystal Clear, DSDM, FDD ja Scaled Agile Framework. (Ambler & Lines 2012, 58.)

DAD viitekehyksen perusajatus on tehdä enemmän kuin ohjelma (software). DAD keskittyy toimittamaan **informaatioteknologiaratkaisun**, joka tuottaa todellista liiketoiminta-arvoa sidosryhmille. (Ambler & Lines 2012, 10.)

DAD tiimi toimii muiden tiimien tavoin yrityksen ekosysteemin puitteissa ja on **yritystietoinen**. Tähän sisältyy läheinen yhteistyö yrityksen teknisten arkkitehtien, liiketoiminta-arkkitehtien, ylemmän johdon, operatiivisen tason edustajien, tietohallinnon, it-tuen, liiketoiminta-asiantuntijoiden, myynti- ja palveluennusteiden, portfolio salkun hoitajan ja muiden tärkeiden tahojen kanssa. DAD tiimit pyrkivät vähentämään tekniikan syvyyttä ja DAD tiimin ajatusten ja toiminnan punainen lanka on ”Jätä se paremmaksi kuin se oli löytäessäsi⁶”. Tosin sanottuna DAD tiimi mukautuu koko organisaation ajattelutapaan. (Ambler & Lines 2012, 17.)

Filosofia ”Hyökkää riskien kimppuun ennen kuin ne hyökkäävät sinun kimppuun” sisältyy DAD viitekehykseen. DAD on omaksunut laajennetun version arvosuuntautuneesta toimintamallista **riski- ja arvosuuntautuneeksi** elinkaareksi. Riski- ja arvosuuntautuneisuus tarkoittaa, että myös suuren riskin sisältävä toiminto saa korkeamman prioriteetin, ei pelkästään suuren arvon tuottava toiminto. DAD:iin sisältyy perustointamallit IT - toimitusten riskien vähentämiseen, kuten mahdollisesti käytettävä ratkaisu, iteraatioiden demot ja aktiivinen sidosryhmien osallistuminen. Lisäksi DAD:iin kuuluu selkeät virstanpylväät, joilla arvioidaan projektin elinkelpoisuutta ja päätetään jatkosta. Nämä virstanpylväät sisältävät sidosryhmien konsensuksen visiosta, aikaisessa vaiheessa todennetun arkkitehtuurin, jatkuvan toteuttamisen kelpoisuuden arvioinnin,

⁶ Mukailtu partioliikkeen perustajan, Robert Baden-Powelin, jäähyväiskirjeestä (Partiowiki 2013)

riittävän toiminnallisuuden, tuotantovalmiuden ja ilahtuneet sidosryhmät. (Ambler & Lines 2012, 19-21.)

DAD tunnistaa **skaalautuvuuden** tarkoittavan enemmän kuin tiimikoon suurentamista. DAD:ssa on monia skaalautuvia osatekijöitä, joita räätälöidään omaan tilanteeseen sopiviksi. Skaalautuvia osatekijöitä ovat maantieteellinen levittyneisyys, tiimikoko, säännönmukaisuuden säätely, liiketoiminnan monimutkaisuus, tekninen monimutkaisuus, organisaation monimutkaisuus, organisaation levinneisyys ja yrityskuri. (Ambler & Lines 2012, 22.)

DAD on **päämääräohjautunut ja toimitukseen keskittynyt elinkaari**. DAD elinkaari käsittää projektin alustuksesta rakentamisen kautta tuotantoon julkaisuun. Projektin elinkaari on leikattu vaiheisiin. Näin varmistetaan, että tehdään oikeita asioita oikeaan aikaan. Kolme vaihetta on alku, rakentaminen ja muutos (kuvio 2). Vaiheet ovat selkeitä ja heijastelevat ketteryuden koordinoi - tee yhteistyössä - saata päätökseen - rytmiä. DAD toimituslinkaari esitellään kontekstissaan ja se tunnistaa tukitarpeen tuotantoon toimituksen jälkeen. DAD sisältää selkeitä virstanpylväitä ja näkee ne luontaisena toimintatapana vähentää hallintoa ja riskejä. (Ambler & Lines 2012, 11.)

4.1 DAD vaiheet ja päämäärät

DAD malliin kuuluu kolme vaihetta, alku-, rakennus- ja siirtymävaihe.

Alkuvaihe on selkeä ja sen tavoitteet ovat muodostaa alustavasti DAD tiimi, tunnistaa projektin visio, saavuttaa osakkaiden yhteisymmärrys visiosta, olla linjassa yrityksen suunnan kanssa, määritellä alustava tekninen strategia, vaatimukset, ja julkaisusuunnitelma, pystyttää työskentely-ympäristö, turvata rahoitus ja tunnistaa riskit. (Ambler & Lines 2012, 111, 113.)

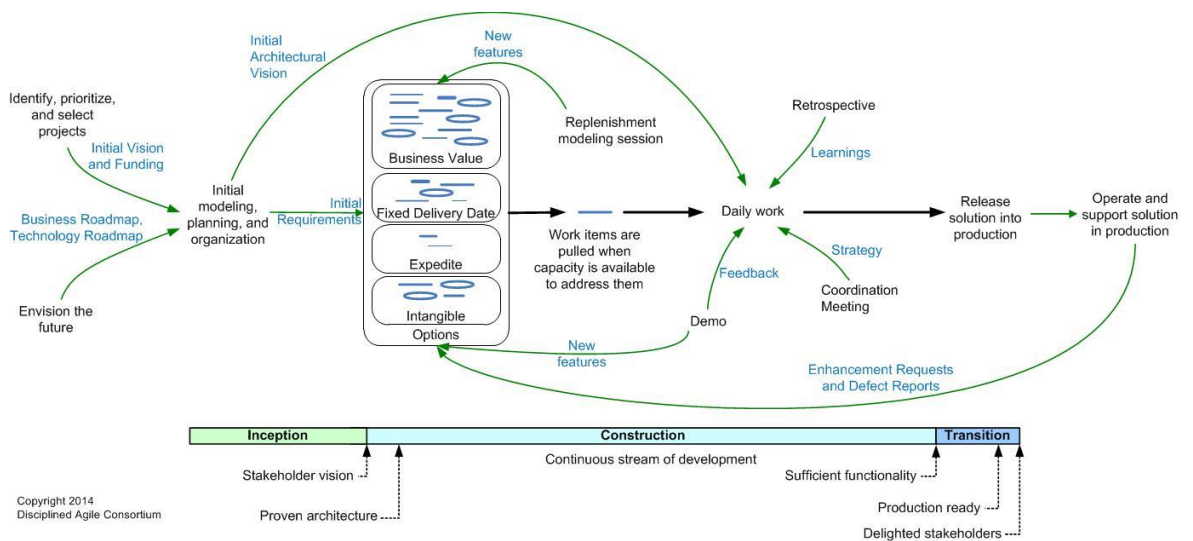
Rakennusvaihe koostuu useista iteraatioista, joissa ratkaisu kehittyy kohti asetettua päämäärää. Ensisijainen rakennusvaiheen tavoite on tuottaa käytettävä ratkaisu, joka tuottaa liiketoiminta-arvoa organisaatiolle ja ylittää tai pysyy toiminnallisuuksien minimivaatimuksissa, jotka oikeuttavat käyttöönoton kustannukset. Rakennusvaiheen päämäärät ovat potentiaalisesti käytettävä ratkaisu, käsitellä osakkaiden muuttuvat tarpeet,

liikkua kohti käyttöönotettavaa julkaisua, ylläpitää tai parantaa laatutasoa ja todentaa arkkitehtuuri varhain. (Ambler & Lines 2012, 273, 275.)

Siirtymävaiheen päätarkoitus on saada käyttöön kulutettava ratkaisu ja toimittaa siihen liittyvät asiakirjat, kuten mm. manuaalit. Siirtymävaiheessa on mahdollisesti useita iteraatioita, jotka ovat kestoaltaan lyhyempiä kuin rakennusvaiheen iteraatiot. Tavoitteita siirtymävaiheelle ovat varmistaa tuotantovalmius, varmistaa osakkaiden valmius vastaanottamaan ratkaisu, käyttöönottaa ratkaisu tuotantoon. (Ambler & Lines 2012, 417-419.)

Näiden kolmen vaiheen päämäärien lisäksi DAD elinkaaren aikana jatkuvasti tavoiteltavia päämääriä ovat täyttää projektin tehtävää, kasvattaa tiimin jäsenen taitoja, parantaa olemassa olevaa infrastruktuuria, parantaa tiimin prosessia ja ympäristöä, hyödyntää olemassa olevaa infrastruktuuria ja käsitellä riskejä (Ambler & Lines 2012, 11).

Kun taito soveltaa DAD:ia lisääntyy, voidaan ajatella DAD:in kevyemmän version käyttämistä (kuvio 3). Kevyemmässä versiossa vaiheita ja iteraatioita lähestytään ”tee se mitä tarvitsee silloin kun se pitää tehdä” - ajatuksella. Tällöinen Lean -tyylinen toimintatapa sopii hyvin vain erittäin kurinalaisille tiimeille. (Ambler & Lines 2012, 11-12.)



Kuvio 3. DAD kevyempi versio (Ambler 2014)

DAD on prosessiviitekehys ja seuraa ketterää 3C rytmiä. Nämä kolme ovat Coordinate (koordinointi), Collaborate (yhteistyö) ja Conclude (päättös). Tämä 3C rytmi esiintyy DAD:ssä kolmella tasolla, julkaisu-, iteraatio- ja päivätaso. Julkaisutasolla ovat edellä esitellyt vaiheet, iteraatiotasolla suunnitellaan, toteutetaan ja päätetään demoön ja retrospektiin ja päivätasolla 3C ilmentyy lyhyellä koordinoitipalaverilla, ratkaisun työstämisellä ja lopuksi toimivalla versiolla. (Ambler & Lines 2012, 14.)

4.2 DAD vastuut ja rooli

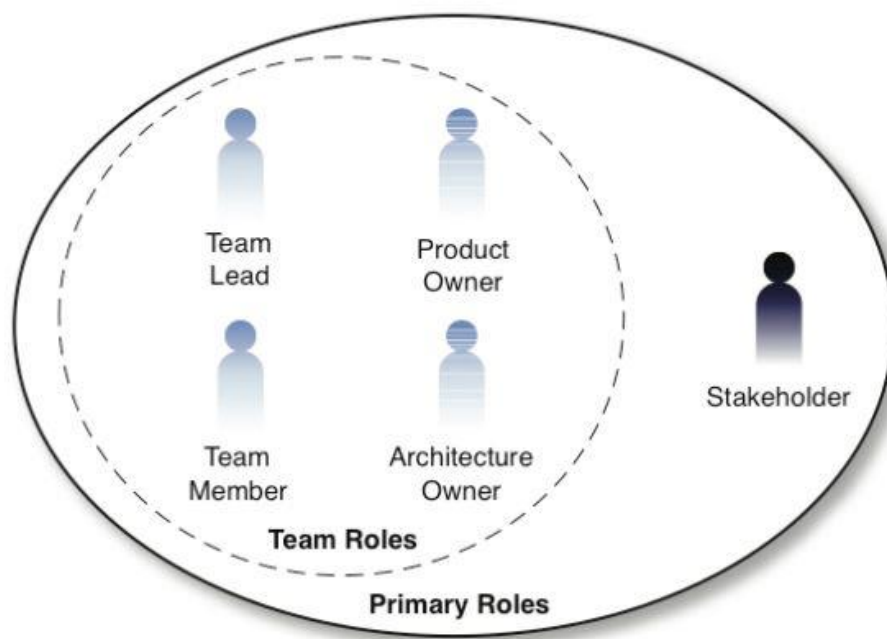
DAD tarkastelee vastuuta oikeuksien kanssa. Kaikilla DAD projektiin osallistuvilla on oikeuksia. Kaikkien oikeuksia ovat tulla kohdelluksi kunnioituksella, oikea-aikainen päätöksen teko, tuottaa ja vastaanottaa laadultaan projektistandardien ja periaatteiden mukaista työtä, arvioida toimintoja, joissa on osallisena ja nauttia muiden arvostusta kyseisistä arvioinneista. Arviointien tulee olla saavutettavissa vaikkakin tiukkoja. Lisäksi kaikkien oikeuksiin kuuluu riittävät resurssit, oikea-aikainen neuvottelumahdollisuus, jos jokin sitoumus näyttää mahdottomalta toteuttaa, päättää omien resurssien käytöstä, saada mahdollisuus kasvattaa tietämystä, joka auttaa projektia menestymään, työskennellä turvallisessa ympäristössä, jossa sallitaan virheet ja nähdään virheet mahdollisuutena oppia, saada kiitosta, saada kasvatusta, saada tukea, saada oikea-aikaisesti oikeaa tietoa, omata organisaation ohjelmistoprosessit, noudattaa ja aktiivisesti parantaa niitä, jos tarvetta ilmenee. (Ambler & Lines 2012, 63-64.)

Edellä mainitut oikeudet ovat helposti käännettävissä myös vastuiksi. Niiden lisäksi jokaisella DAD tiimin jäsenellä on vastuu tuottaa paras mahdollinen ratkaisu sidosryhmien tarpeisiin annetuilla resursseilla, optimoida annettujen resurssien käyttö, työskennellä vapaaehtoisesti tiimin kesken myös oman erikoisosaamisalueen ulkopuolella, jakaa tietoa, jakaa tietoa työnalla olevasta työstä, valmentaa muita oman taidon ja osaamisen osalta, tarkistaa aikaisessa vaiheessa, että tekee oikeanlaista työtä, aktiivisesti laajentaa omaa tietämystä ja osaamista ja käyttää osaamista oman erikoisosaamisalueen ulkopuolella tarvittaessa, osallistua henkilökohtaisesti koordinoituihin palavereihin, jos paikalla tai osallistua muulla tavalla, edesauttaa parantamaan oman ja tiimin suoritusta koko pro-

jektin ajan, välttää hyväksymästä työtä, joka on käsillä olevan iteraation ulkopuolella. (Ambler & Lines 2012, 64-65.)

DAD roolit eivät ole positioita, työpaikkoja, eikä niiden ole tarkoituskaan olla. Esimerkiksi henkilö X on organisaatiossa positiossa rahoituspäällikkö ja projektissa sidosryhmän edustaja. Projektissa saattaa olla monia sidosryhmän edustajia, ja on epätodennäköistä, että kenenkään työpaikkarooli on sidosryhmän edustaja. (Ambler & Lines 2012, 65.)

Ensisijaiset roolit ovat löydettävissä DAD projektissa aina laajuuteen tai monimutkaisuuden tasoon katsomatta (kuvio 4). Roolinimet ensisijaisille rooleille ovat Sidosryhmän edustaja ts. osakas ja tiimin roolinimet ovat Tiimin jäsen, Tiimin johto, Tuoteomistaja ja Arkkitehtuuriomistaja. (Ambler & Lines 2012, 65-66.)



Kuvio 4. Ensisijaiset DAD roolit (Ambler 2012)

Osakas rooli on jaoteltu neljään kategoriaan, jotka ovat loppukäyttäjät, päämiehet, partnerit ja sisäpiiriläiset. Loppukäyttäjät odottavat järjestelmän olevan käytettävä ja mahdollistavan heidän työnsä tehokkaamman tekemisen. Päämiehet tekevät päätökset, maksavat järjestelmästä ja ottavat järjestelmän käyttöön. Partnerit huolehtivat järjestelmän tuotantoon ottamisesta. Sisäpiiriläiset ovat kehitystiimin jäsenet, joihin kuuluvat

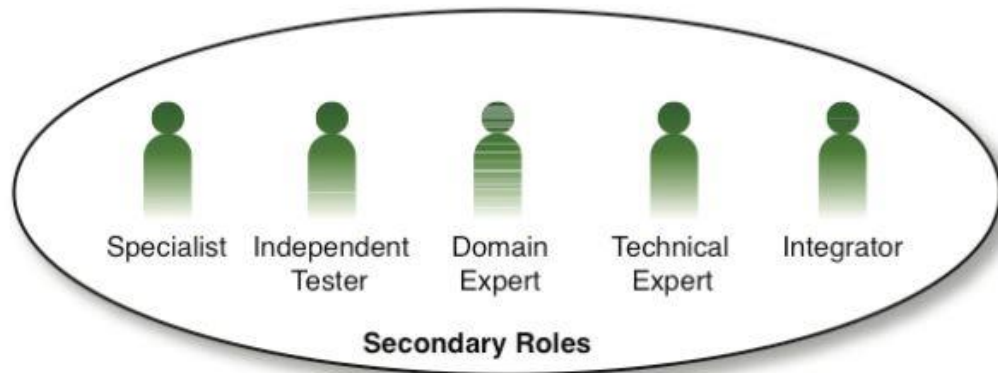
mm. DBA, tietoverkkoasiantuntija, markkinointi asiantuntija ja myyjät. (Ambler & Lines 2012, 67.)

Tuoteomistaja on osakkaiden ääni DAD tiimissä. Tuoteomistaja antaa vastaukset kysymysten ratkaisua ja tuoteomistaja ylläpitää priorisoitua työtehtävälistaa. Toissijaisia tuoteomistajan tehtäviä ovat demotilaisuuksien järjestäminen ja avainosakkaiden kanssa neuvottelu ja kommunikaatio. (Ambler & Lines 2012, 68-69.)

Tiimiläiset vastaavat ohjelmoinnista, testauksesta, suunnittelusta, arvioinnista ja analyysistä. Jokaisen ei tarvitse osata jokaista tehtäväkenttää projektin alussa, mutta jokainen oppii per projekti jotain uutta omien vahvuuksien ulkopuolelta. Tiimissä on myös vetäjä, joka on tiimin mahdollistaja ja ketteryysvalmentaja. Hän varmistaa tiimin tehokkaan työskentelyn ja jatkuvan kehittymisen paremmaksi. (Ambler & Lines 2012, 71 & 73.)

Arkkitehtuuriomistaja tekee arkkitehtuuriin liittyvät päätökset tiimille ja mahdollistaa ylipäättään ratkaisun luomisen ja kehittymisen. Tehtäessä pienempiä ratkaisuja on mahdollista yhdistää arkkitehtuuriomistajan ja tiimin vetäjän vastuut. Arkkitehtuuriomistaja suorittaa tiimin jäsenen toimia siinä missä muutkin tiimiläiset. (Ambler & Lines 2012, 76.)

Toissijaiset roolit astuvat kuvaan usein projektin laajetessa. Nämä toissijaiset (Kuvio 5) roolit ovat usein väliaikaisia eikä jokaista roolia usein tarvitse täyttää jokaisessa projektissa. Mutta on todennäköistä, että jotain toissijaisia rooleja tarvitaan jokaisessa projektissa. Toissijaisten roolien edustajat omaavat erityistietämystä esim. tekniikasta tai toimialasta. Toissijaisia rooleja ovat mm. toimiala-asiantuntija, tekniikka-asiantuntija, spesialisti, itsenäinen testaaja ja integraattori. (Ambler & Lines 2012, 65, 78.)



Kuvio 5. DAD toissijaiset roolit (Ambler 2012)

Toimiala-asiantuntijan erityisosaamista tarvitaan tiimiin esimerkiksi verotusasioihin liittyvien vaatimusten selventämiseksi. Spesialistia tarvitaan suurissa tiimeissä tai monimutkaisissa toimialoissa tuomaan tietämystä tiimeihin tiimin vetäjän kautta. Spesialisteja voidaan nähdä myös aloittelevissa DAD tiimeissä. Tekniikka-asiantuntija tulee vastaamaan erikoistarpeisiin kuten mm. käyttöliittymäsuunnittelu, ketterä tietokanta-asiantuntija. Tekniikka-asiantuntijaa tarvitaan hetkellisesti ja hän mahdollistaa nopean ratkaisun ongelmaan. Itsenäinen testaajatiimi työskentelee rinnakkain DAD tiimin kanssa validoiden heidän työtään koko projektin elinkaaren ajan. Integraattori astuu kuvaan, kun iso DAD tiimi on jaettu pienempiin alitiimeihin. Tällöin integraattori vastuulla on alitiimien työn tulosten integroiminen kokonaisuudeksi. (Ambler & Lines 2012, 78-79.)

4.3 DAD arvojen ja periaatteiden yhteneväisyydet ja erot ketteriin periaatteisiin

Agile Manifeston arvoja kannattaa ajatella suosituksina ja painotuksina, ei vaihtoehtoina, rohkaisuna keskittää huomio tiettyihin alueisiin, mutta ei pois sulkea muita. Discip-

lined Agile Values eroavat Agile Manifeston osalta (Ambler & Lines 2012, 27-29) seuraavasti:

- *Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja.*

Erot: Ei ole, DAD on täysin omaksunut tämän arvon.

- *Toimivaa ratkaisua enemmän kuin kattavaa dokumentaatiota.*

Erot: Ohjelmiston sijaan DAD painottaa toimivaa ratkaisua. Ratkaisussa otetaan huomioon, ei vain ohjelmisto vaan, mahdolliset laitteistomuutokset (hardware), liiketoimintaprosessien tuki ja dokumentaatio siitä ja ratkaisun kanssa työskentelevät ihmiset.

- *Sidosryhmäyhteistyötä enemmän kuin sopimusneuvotteluja.*

Erot: Asiakasyhteistyö laajennetaan koskemaan sidosryhmiä. Selkeästi tunnustetaan, että ratkaisuun liittyviä sidosryhmiä on laajalti eikä ainoastaan yritysasiakas ja että tarvitaan yhteistyötä myös muiden kuin sidosryhmän edustajan kanssa, jotta päästään ymmärrykseen todellisista tarpeista.

- *Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa.*

Erot: Ei ole.

Agile Alliancen jäsenet jalostivat julistuksen 12:teen periaatteeseen. Näiden periaatteiden tarkoitus on auttaa ymmärtämään mitä ketterä ohjelmistokehitys on. DAD:ssä on muokattu monia näistä periaatteista ja lisäksi on syntynyt kolme uutta periaatetta. (Ambler & Lines 2012, 29-32.)

1. Tärkein tavoitteemme on tyydyttää sidosryhmät toimittamalla tämän tarpeet täyttäviä versioita ratkaisusta aikaisessa vaiheessa ja säännöllisesti.

Erot: Laajennettu käsitettä, kenelle tehdään ratkaisua, ei vain ohjelmistoa, ei vain asiakkaalle vaan kaikille sidosryhmille.

2. Otamme vastaan muuttuvat vaatimukset myös ratkaisun toimituksen myöhäisessä vaiheessa. Ketterät menetelmät hyödyntävät muutosta sidosryhmien kilpailukykyyn edistämiseksi.

Erot: Painotetaan ratkaisua, joka toimitetaan ohjelmistokehitysprosessin sijaan. Tavoitteena on tuottaa hyötyä sidosryhmille kokonaisuudessa.

3. Toimitamme versioita toimivasta ratkaisusta säännöllisesti, parin viikon tai kuukauden välein, ja suosittelemme lyhyempää aikaväliä.

Erot: Laajennettu käsite lopputuotteesta, ei vain ohjelmisto vaan ratkaisu.

4. Sidosryhmien edustajien ja ohjelmistokehittäjien tulee työskennellä yhdessä päivittäin koko projektin ajan.

Erot: Painotetaan sidosryhmien osallistumisen tärkeyttä. Käytetään työkaluja, jotta osallistuminen mahdollistuu.

5. *Rakennamme projektit motivoituneiden yksilöiden ympärille. Annamme heille puitteet ja tuen, jonka he tarvitsevat ja luotamme siihen, että he saavat työn tehtyä.*

Erot: Ei ole.

6. Tehokkain ja toimivin tapa tiedon välittämiseksi toimitustiimille⁷ ja tiimin jäsenten kesken on kasvokkain käytävä keskustelu.

Erot: DAD ei keskity pelkästään kehitystiimiin vaan ratkaisu toimitetaan ja tehdään toimitustiimin voimin.

7. Laskennallinen liiketoiminta-arvo on edistymisen ensisijainen mittari.

Erot: DAD painottaa arvon tuottamista, ei pelkästään toimivaa ohjelmistoa, mittarina edistymisestä.

8. Ketterät menetelmät kannustavat kestävään toimitustapaan. Hankkeen omistajien, kehittäjien ja ohjelmiston käyttäjien tulisi pystyä ylläpitämään työtahtinsa hamaan tulevaisuuteen.

Erot: DAD:ssa on laajennettu toimintatapa käsittämään koko toimitustapaa.

9. *Teknisen laadun ja ohjelmiston hyvän rakenteen jatkuva huomiointi edesauttaa ketteryyttä.*

Erot: Ei ole.

10. *Yksinkertaisuus - tekemättä jätettävän työn maksimointi - on oleellista.*

Erot: Ei ole.

11. *Parhaat arkkitehtuurit, vaatimukset ja suunnitelmat syntyvät itseorganisoiduissa tiimeissä.*

Erot: Ei ole.

12. *Tiimi tarkastelee säännöllisesti, kuinka parantaa tehokkuuttaan, ja mukauttaa toimintaansa sen mukaisesti.*

Erot: Ei ole.

⁷ Suomennettu delivery team sanasta.

13. Käytä ja kehitä organisaation voimavaroja yhteistyössä niiden ihmisten kanssa, jotka ovat vastuussa niistä voimavaroista.
- Erot: Uusi DAD periaate. Tarkoittaa, että DAD tiimit tiedostavat etteivät toimi tyhjiössä vaan edistävät olemassa olevan infrastruktuurin käyttöä ja parantavat sitä.
14. Visualisoi työnkulku saavuttaaksesi toimitukselle jouheva työnkulku samalla piten työnalla oleva tehtävämäärä minimissään.
- Erot: Uusi DAD periaate. Visualisoimalla työtehtävien status ja rajoittamalla WIP määrää vähennetään pullonkauloja työstä.
15. Organisaation tulee kehittyä heijastamaan ja edistämään ketterien tiimien pyrkiä olemalla kuitenkin riittävän joustavia ei-ketteriä ja hybriditiimejä kohtaan.
- Erot: Uusi DAD periaate. Kokonaisvaltaisen IT strategian tulee heijastua yksittäisiin tiimeihin, jotta niiden yhteistyö on mahdollisimman tehokasta.

5 Tutkimuksen toteutus, aineisto ja tulokset

Tutkimus on kaksiosainen. Tutkimuksen ensimmäinen osa tehdään empiirisenä tutkimuksena 2006 tai sen jälkeen julkaistusta opinnäytetyöstä ja yliopistotutkimuksesta, joiden tutkimuskohteena on ollut ketterän sovelluskehityksen ja perinteisen projektinhallinnan yhdistäminen. Aineistona käytettiin lisäksi vuonna 2013 julkaistua pro gradu tutkielmaa, jossa tutkittiin ketterien metodologioiden perinteisen projektinhallinnan ominaisuuksia sovelluskehityksessä. Aineistona käytettiin myös tekijän työharjoittelussa syksyn 2013 aikana tehtyjä havaintoja.

Toisessa osassa tutkitaan vastaako DAD perinteisen ja ketterän projektinhallinnan yhdistämisessä löydettyihin ongelmiin sovelluskehityksessä. DAD viitekehys otettiin mukaan opinnäytetyöhön, koska tuntui järkeenkäyvältä tutkia ensin onko ajankohtainen prosessiviitekehys mahdollisesti toimiva ennen kuin lähtee kehittämään uutta mallia.

Tutkimusaineiston valinta tapahtui ensisijaisesti tekijän oman kiinnostuksen pohjalta. Työharjoittelun aikaiset tapahtumat jäivät askarruttamaan mieltä siinä määrin, että tunsin tarvetta tutkia mahdollisia yhteneväisyyksiä muihin vastaavanlaisiin projekteihin. Perinteisen ja ketterän projektinhallinnan yhdistämisestä on saatavilla tietoa. Tosin tietoa siitä, onko jo tiedossa olevista yhdistelmämallista vastaamaan ohjelmistokehitysprojektiin koko sen elinkaaren ajan, ei ollut varmuutta. Aineiston julkaisun aikaväli rajoittui automaattisesti 2000 -luvun jälkeiseen aikaan, koska ketterä kehitys, on jalkautunut projektityöhön 2001 annetun Agile Manifeston jälkeen.

5.1 Aineiston esittely

Ensimmäinen tutkimus on nimeltään Projektinhallintamenetelmien PM methodology ja Scrum yhdistäminen (Solanterä 2010) ja se tutkii suuren organisaation käyttämän perinteisen projektinhallinnan viitekehityksen yhdistämistä ketterään viitekehitykseen kolmessa eri kokoluokan projektissa. Organisaatiossa on ollut käytössä projektinhallintamalli, jossa selkeät vaiheet seuraavat toisiaan erilaisten tarkastuspisteiden välityksellä. Organisaatiossa yksittäiset tiimit olivat kokeilleet ketterien käytänteiden yhdistämistä perinteiseen projektinhallintaan. Näiden kokeilujen tulokset olivat rohkaisevia ja orga-

nisaatiossa keskijohdon kannustuksella alettiin etsiä erikokoisiin ja erilaisilla kriittisiin projekteihin sopivaa yhdistelmämallia. Tutkimus suoritettiin projektijohtoa haastattelemalla.

Toisessa tutkimuksessa Karlström ja Runeson (2006) tutkivat kahden suuren organisaation yhden projektin tuloksia integroitaessa ketteriä menetelmiä perinteiseen projektinhallintaan. Molemmissa organisaatioissa oli pitkät perinteet ohjelmistokehitykseen ja projektinhallintaan. Noudatellut perinteistä mallia jo vuosia. Tutkimuksen aineiston organisaatioista toisessa oli johdon tuki ketterien käytänteiden käyttöönotolle ja ketterien menetelmien käyttöönotto mahdollistettiin ulkoisen konsultin avulla ja toimesta. Toisessa organisaatiossa käyttöönoton mahdollistivat ohjelmistokehittäjät ja johdon tuki uusille käytänteille oli vähäistä. Tutkimus suoritettiin haastattelemalla ohjelmistokehittäjiä ja keskijohtoa.

Kolmas, käsitteellis-teoreettinen, pro-gradu tutkielma on nimeltään Projektinhallinta ketterässä sovelluskehityksessä (Penttonen 2013) ja siinä tutkitaan ketterien metodologioiden projektinhallintaominaisuuksia. Tutkimuksessa perehdytään kolmeen ketterään viitekehykseen ja pyritään vastaamaan kysymykseen, mitä projektinhallinnallisia ominaisuuksia ketterät viitekehykset sisältävät ja vastaavatko ne projektinhallinnan tarpeisiin koko projektin elinkaaren ajan.

Neljäs tutkittu aineisto perustuu opinnäytetyön tekijän havaitsemiin asioihin omassa työharjoittelussaan, jossa ketteryys ja perinteinen projektimalli yhdistettiin. Nämä havainnot tehdään asiakkaan näkökulmasta. Yrityksessä oli meneillään laaja toiminnanohjausjärjestelmän uudelleen rakennus ja laajennus. Uudistusta toteutettiin kolmen tiimin kesken. Tiimeillä oli valtavia osaamiseroja niin projektinhallinnassa kuin sovelluskehityksessä.

5.2 Tulokset ketterän ohjelmistokehityksen ja perinteisen projektinhallinnan yhdistämisen ongelmista

Aineiston perusteella voidaan sanoa, että perinteinen projektinhallinta ja ketterä ohjelmistokehitys on mahdollista yhdistää ja sen suuntaista kehitystä toivotaankin organisaatioissa, mutta ongelmatonta se ei ole. Ketteryyden koetaan olevan pääasiallisesti toteutusvaiheen viitekehys ja lopputuloksena yhdistämisessä on scrummenfall hyötyä tuottavan yhdistelmän sijaan. Tällöin kadotetaan ketteryyden hyödyt, koska lopputuloksena on jäykkää projektitoimintaa.

Ketteryyden yhdistämistä perinteiseen projektinhallintamalliin ei koettu missään helpoksi. Päinvastoin, yhteensovittaminen koettiin ongelmaksi muun muassa yhdistettäessä ketterä osaprojekti perinteiseen summaprojektiin ja tehtäessä ja seurattaessa summaprojektin kustannusarviota. Ongelmana oli tehdä tarpeeksi tarkkoja laskelmia esim. tarinakorttiin perustuen. Projektien budjetoiminen suurpiirteisesti tai liian tiukasti antaa epärealistisen kuvan johdolle projektin mahdollisuuksista ja aloituksen jälkeen projektin tilasta.

Yhteensovittaminen koettiin vaikeaksi myös suunniteltaessa iteraatioiden julkaisujen aikataulua. Ongelmana oli sovittaa ne perinteisen mallin merkkipaalujen välille. Merkkipaalujen tai porttivaiheiden kautta seuraavaan vaiheeseen siirtyminen koettiin ongelmaksi tilanteessa, jossa ketterä tiimi olisi ollut valmis aloittamaan, mutta ei-ketterä tiimi odotti selkeää dokumenttia vaatimusmäärittelyjen valmistumisesta ja lupaa aloittaa. Eli toisin sanoen ketterien tiimien ja ei-ketterien tiimien työskentelyn synkronointi on haasteellista.

Osaaminen koettiin lähes jokaisessa tutkimuksessa ongelmaksi. Osaamisen puute ilmentyi pääasiallisesti keskijohdossa ja ylemmässä johdossa. Osaamista ja tietämystä oli yleisellä tasolla, mutta yhdistämiseen ei ollut tarpeeksi tietämystä eikä kokemusta. Osaamisen puute voitiin havaita myös sisältöosaamisessa. Esimiestasolla ei ymmärretty tarvittavan informaation olevan epäformaaleissa dokumenteissa. Ohjelmistokehitysprojekteissa projektipäällikkö joutui tilanteisiin, joissa jouduttiin aikaisessa vaiheessa otta-

maan kantaa tekniseen yksityiskohtaan ja ei-tekniinen projektipäällikön osaaminen ei välttämättä riitä kommentin antamiseen.

Asenne oli yksi ongelmatekijä. Keskijohto ja ylempi johto olivat eniten vastahakoisia uusien käytänteiden implementoimista kohtaan. Yleisesti muutosvastarinta havaittiin olevan yhteinen ongelmakohhta yhdistämisessä. Se että johdossa oltiin vastahakoisia ottamaan erilaisia käytäntöjä prosessiin mukaan, aiheutti ongelmia ketteryyden täysipainoiselle ja -määräiselle yhdistämiselle. Tämä ilmeni niin, että johto pyrki keskittymään makrosuunnitteluun ja raportteihin sen sijaan, että olisivat keskittyneet käsillä olevan työn yksityiskohtiin.

Aineistosta ilmeni käyttöönoton tapaan liittyvä ongelma. Kehittäjät pääsääntöisesti ovat innokkaimpia ketterien menetelmien käyttöönottoon. Alhaalta ylös tapahtuva prosessi aiheuttaa organisaatioissa kitkaa, koska johto on tottunut olemaan kontrollissa organisaation uusista prosesseista ja sen vuoksi kavahtavat uudenlaista tapaa toimia, joka tulee heidän tietoisuuteen komentoketjun alapäästä. Alhaalta - ylös käyttöönotto heikensi johdon ja esimiesten motivaatiota. Esimiesten alhainen motivaatio vaikutti koko projektitiimin asenteisiin ja sitoutumisasteeseen.

Kommunikaatio oli yksi suurimmista ongelmista. Kommunikointiongelmia havaittiin niin tiimien välissä kuin johdon ja tiimien välissäkin. Ketterien tiimien sisällä tapahtuvaa kommunikaatiota ei kyetty tuomaan koko projektinhallintaan yhdeksi käytännöksi. Tämä aiheutti informaatiokatkon, joka puolestaan aikaan sai yleisen epätietoisuuden tilanteesta. Epätietoisuus puolestaan synnytti epäilyjä johdon mieliin.

Tiimien välille muodostui ongelmia kommunikaatioon silloin kun tiimien ketteryyden tasoerot kasvoivat.

Vastuiden hahmottaminen entuudestaan tuntemattomien roolinimikkeiden välillä havaittiin ongelmaksi. Ongelmatilanne havaittiin, kun projektityöntekijä ei enää tiennyt kuka vastasi ja mistä. Vastuiden ja roolinimien hahmottamisvaikeus sekoitti projektiorganisaatiota entuudestaan ja osaltaan lisäsi epätietoisuutta kuka vastaa mistäkin asiassa. Roolinimikkeiden ja niihin liittyvien vastuiden aiheutti projektin product

ownerin siirtämisen takaisin päivittäisiin tehtäviin. Päätöksen teki linjaorganisaation esimies, jolla ei ollut käsitystä projektin rooleista ja vastuista.

Kaikkien edellä mainittujen ongelmien lisäksi ja niihin sisältyen tutkimusaineiston perusteella voidaan lisätä, että ongelmia aiheutti resursointi. Ei osattu antaa ohjelmistokehitysprojektille riittävästi resursseja tai ohjattiin henkilö tekemään työtä, joka ei edistänyt projektia.

Asiakkaan osallistuminen oli ongelma. Asiakas osallistuu liian vähän projektitiimin kanssa yhdessä tapahtumiin. Tämä aiheutti kehitystyön hidastumaa ja turhautumista kehitystiimissä.

Sopivan persoonan ja osaamisen löytäminen projektitiimiin koettiin haasteelliseksi. Esimerkiksi ketteröitymisen alkuvaiheessa olevan organisaation edusta saattaa suhtautua epäilevästi suunnitteluvaiheessa ketterän toimittajan ehdotuksiin. Tarvitaan erinomaisia neuvottelutaitoja, jotta vältetään pattitilanteelta, jossa ei päästä eteenpäin realistisen ja toteutettavissa olevan työmäärän ja toiveiden kanssa. Oikean persoonan ja osaamisen löytäminen vähentää myös epäpätevyyden tuomaa yleisen työmotivaation laskun uhkaa.

Projektinhallinnalliselta kannalta katsottuna aineiston perustella voidaan todeta, että ketterien menetelmien käytännöt ovat kokonaisuuden kannalta ongelmallisia riskien ja vaatimusten hallinnan näkökulmista. Vaatimusten hallinta tapahtuu perinteisissä malleissa pääasiallisesti suunnitteluvaiheessa. Tällöin ketteryyttä ei päästä hyödyntämään täysimääräisesti vaan päädytään ns. scrummenfall tilanteeseen, jossa ketterä ohjelmistokehitysvaihe sisällytetään esim. vesiputousmallin yhdeksi vaiheeksi.

Kokonaisarkkitehtuurisuunnitelman puuttuminen ketterissä projekteissa nousi ongelmaksi. Arkkitehtuuria joko rakennetaan inkrementti kerrallaan tai se nojaa tiimin luomaan metaforaan. Arkkitehtuurin rakentaminen tapahtuu epämuodollisten keskusteluiden ja kasvokkain tapahtuvan kommunikation perusteella.

5.3 Vastaako DAD löydettyihin ongelmiin

Yhdistämisessä löydettiin ongelmia ja haasteita resursoinnissa, asiakkaan osallistumisessa, oikean persoonan löytämisessä ja vastuuttamisessa, vaatimusmäärittelyssä, asenteissa, kustannus- ja aikatauluarvioinnissa, projektinhallintamallin sisältöosaamisessa, käytönotossa, riskienhallinnassa, kommunikaatiossa ja kokonaisarkkitehtuurissa.

Resursointi turvataan aloitusvaiheessa alustavan DAD tiimin kasaamisella. Tämä tiimi kehittyy koko projektin elinkaaren ajan, koska kaikkia ei tarvita projektin ensimmäisestä päivästä alkaen. Alustavalla DAD tiimillä tulee olla tarvittava osaaminen, jotta työ saadaan tehdyksi. Tällä alustavalla tiimillä tulee olla mahdollisuus keskustella osakkaiden kanssa, jotta projektin visio pystytään kirkastamaan ja tiimillä on mahdollisuus etsiä oikeita jäseniä tiimiinsä. Näin varmistetaan riittävä resursointi siirryttäessä seuraavaan vaiheeseen.

Oikean persoonan löytyminen helpottuu DAD:ssä alkuvaiheessa. Tällöin perustetaan alustava tiimi ja tässä vaiheessa voidaan jo vaikuttaa oikeiden persoonien löytämiseen. Tämä ei ole pomminvarma, mutta se kasvattaa todennäköisyyttä, että löydetään oikea persoona rooliin.

Vaatimusmäärittelyongelma käsitellään DAD:ssä aloitusvaiheesta alkaen. Aloitusvaiheessa tehtävä alustava vaatimusmäärittely edistää tuottavuutta, koska perustavanlaiset liiketoimintakysymykset ovat avattu. Lisäksi se, että vaatimuksia identifioidaan varhaisessa vaiheessa, vähentää liiketoimintariskiä ja lisää ratkaisun skaalautuvuutta. Rakenusvaiheessa ratkaisua kehitetään useissa iteraatioissa ja DAD prosessiin kuuluu vastata muuttuneisiin tarpeisiin ja vaatimuksiin.

Asiakkaan osallistuminen on varmistettu DAD:in aloitusvaiheessa ja vaiheesta alkaen. Tämän vaiheen yksi päätavoite on muodostaa alustava DAD tiimi, joka puolestaan visioi osakkaiden kanssa projektin vision. Alustusvaiheen lopussa on saavutettu osakkaiden konsensus.

Kielteiset asenteet organisaatiolle uusia käytänteitä kohtaan ovat DAD:ssä huomioitu samoin kuin ketterässä ajattelutavassa. Ketteryys on ajattelutapa, jonka myötä ketteryyttä harjoittavat tuntevat itsensä enemmän voimaantuneiksi, sitoutuneimmiksi ja sitä myötä asenteet muuttuvat myönteisimmiksi.

Kustannus- ja aikatauluarviointi ovat ongelmia yhdistettäessä ketteryys ja perinteinen projektinhallinta ohjelmistokehityksessä. DAD vastaa rahoitusta mietittäessä aloitusvaiheessa turvaamalla rahoituksen. Tähän voidaan käyttää erilaisia strategioita, kuten fixed price, stage gate tai time and materials. Aloitusvaiheessa suoritetaan myös alustavan julkaisuaikataulun suunnittelu. Alustavanjulkaisuaikataulun suunnittelulla pyritään kohdentamaan ajatuksia siihen, mitä tehdään sen sijaan, että suunnitellaan semmoista, mitä luullaan tehtävän.

Mallin käyttöönotto, osaaminen ja käyttö ovat DAD viitekehyksessä huomioitu vaihe vaiheelta. DAD:in käyttöönoton onnistumista ja sitoutumista uuteen prosessiviitekehykseen pohjustetaan läpi organisaation mm. lausumalla julki DAD tiimien olevan yritystietoisia. Tämä yritystietoisuus auttaa jokaisella organisaatiotasolla omaksumaan uuden ajattelutavan sen sijaan, että pieni kehittäjäryhmä puskisi uutta tapaa toimia ja ajatella alhaalta ylöspäin. Käyttöönottoon on myös julkaistu omia white paper oppaita viitekehyksen isien kirjoittamana. Osaaminen ja viitekehyksen soveltaminen on varmistettu suosimalla coaching -lähetysmäistapaa. Tässä tavassa osaaminen kehittyy ja soveltaminen helpottuu, kun DAD tiimit käyttävät kokeneita agilesteja toimintansa kehittämisen apuna parantaessaan oppimistaan. Tällä tavoin alleviivataan DAD:in yhtä osaa, oppimissuuntautumista.

Riskienhallinta sisältyy jatkuvana toimintana DAD prosessiin. Riskien hallintaa tarkastellaan samoin kuin toimitettavan ratkaisun arvoakin. Myös todennäköiset ja suurta vahinkoa tuottavat riskit huomioidaan jatkuvasti antamalla niille korkea prioriteetti work items listalle.

Kasvokkain tapahtuva kommunikaatio lisättynä havainnollistavilla piirroksilla on todettu tehokkaimmaksi tavaksi välittää tietoa. Tutkimusten mukaan ketterän tiimin sisällä tapahtuvaa kommunikaatiota ei pystytty tuomaan muulle organisaatiotasolle ja ei-

ketterien tiimien käyttöön riittävän tehokkaasti. Tämä ongelma on huomioitu DAD viitekehysessä, mm. DAD periaatteissa on kirjattu, että live kommunikaatiota tulee tapahtua DAD projektin koko elinkaaren ajan päivittäin. Fyysisesti eri lokaatioissa olevien projektihenkilöiden tulee mahdollistaa kommunikaatio valitsemalla eri vaihtoehtoista tehokkaimman tavan, kuten esimerkiksi webkameran avulla suoritettavan videoneuvottelun. Ja DAD:in oppimissuuntautuneisuus periaate vahvistaa käytänteiden jatkuvaa parantamista.

Kokonaisarkkitehtuuri huomioidaan DAD:ssä aloitusvaiheesta alkaen. Aloitusvaiheessa määritellään alustava tekninen strategia. Rakennusvaiheen alkumetreillä todistetaan arkkitehtuuri toimivalla koodilla. Arkkitehtuuria kehitetään koko projektin elinkaaren ajan ja arkkitehtuuriomistajan vastuulle kuuluu ohjata ja kehittää tiimiä ymmärtämään arkkitehtuuria.

6 Yhteenveto

Opinnäytetyön tavoitteena oli aluksi tutkia ketteryiden ja perinteisen projektinhallinnan yhdistämisen ongelmia ja sen jälkeen tutkia vastaako DAD löydettyihin ongelmiin. Aineistoksi pyrittiin löytämään tapauksia, joissa tarkasteltiin laaja-alaisesti projektia. Ulkopuolelle jätettiin vain yhdestä näkökulmasta, kuten ainoastaan riskienhallinnan tai yhden sidosryhmän, tarkastellut työt.

Kriittisiksi ongelmiksi nousivat selkeästi osaaminen, johdon asenne, ketteryiden alhaalta ylös käyttöönottotapa ja tasapainoilu sen välissä, kuinka ketteriä ollaan ja kuinka ketteruus yhdistetään perinteisiin vaiheisiin ja merkkipaaluihin tai kuinka ketteryydeltään eritasoiset toimijat synkronoivat työnsä. Osaamisen puutteen vuoksi ketterien menetelmien käytänteet eivät pystyneet tuomaan hyötyä projektille. Johdon asenne ongelma ja käyttöönoton tapa ongelma liittyivät osaksi toisiinsa. Johdossa ei osattu tai haluttu ottaa käyttöön alemmalta tasolta ehdotettuja työtapoja. Toisaalta ketterien menetelmien mukaisesti saattoi projektipäällikkö joutua tilanteeseen, jossa hänen tuli ottaa kantaa tekniseen yksityiskohtaan. Ketterien ja perinteisten menetelmien välillä tasapainoilu oli ongelmallista. Koska osaaminen oli yksi ongelmista, päädyttiin scrummemfalliin, jossa ketterät käytänteet ovat vesiputousmallin yksi vaihe. Tiimien ketteryyserojen ollessa suuria, ketterän tiimin osaamista ei osattu tuoda ulos tiimin sisältä ja se aiheutti töiden synkronointiin eriaikaisuutta ja herätti kysymyksiä johdossa projektin tilasta. Synkronointiongelma aiheutti aikaresurssin tuhlaamista ja johdon tietämättömyys johti huonoihin päätöksiin resursoinnissa. Ongelmat aiheuttivat myös projektien ketteryyden hyötyjen vajavaisen hyödyntämisen. Käytännössä kaikki ongelmat juontavat juurensa ketteryyden sisäistämisessä. Usein toimija ajattelee toimivansa ketterästi jättämällä dokumentaation ja suunnittelun pois. Agile Manifeston periaatteiden ja aatteiden sisäistäminen ja käyttöönotto vaatii aikaa, harjoitusta - kukaan ei ole ketterä syntyessään - ja lisäksi kokeneiden agilistien käyttöä ketteryyden alkutaipaleella suositellaan.

DAD on prosessiviitekehys ja se eroaa ketterästä laajentamalla ja syventämällä ketterää ajattelu- ja toimintatapaa. DAD ottaa huomioon organisaation ja tunnustaa kaiken perustana olevan organisaation strategia ja halu parantaa prosesseja läpi organisaation. DAD puhuu ratkaisusta ohjelmiston sijaan ja haluaa näin korostaa tietotekniikan roolia

arvon tuoton mahdollistajana organisaatiolle. DAD ajattelee toimivan ohjelmiston olevan riittämätön, ohjelmisto voi olla toimiva, mutta ei vastaa tarpeisiin.

DAD vastaa perinteisen projektinhallinnan ja ketteryyden yhdistämisessä löydettyihin ongelmiin. DAD vaatii kuitenkin harjoittelua, valmennusta ja opettelua. DAD nojaa valtaosaltaan yli vuosikymmenen takaiseen ketteryyden julistukseen, joten voidaan hyvin kysyä, riittäisikö ketteryyden sisäistäminen ja opettelu ratkaisemaan ongelmat.

Opinnäytetyön tavoitteet saavutettiin hyvin. Alkuoletuksena lähtiessäni tekemään tutkimusta, minulla oli oletuksena löytää samoja ongelmia kuin oman työharjoittelujakson perusteella havaitut ongelmat. Oli mielenkiintoista havaita yhteneväisiä ongelmia muiden aineiston tapauksista. Ensimmäisessä vaiheessa tavoitteeni oli rakentaa prosessimalli, jonka avulla pystytään vastamaan löydettyihin ongelmiin ja päästyäni aiheeseen kunnolla sisään, löysin ajankohtaisen prosessimallin, DAD:in, ja päätin ottaa sen mukaan tutkimukseen. DAD:in kehittämiseen johtaneet haasteet ketteryyden tuomisessa organisaatioihin ovat globaaleja ja ajankohtaisia, joten opinnäytetyöprosessi opetti minua loppujen lopuksi enemmän kuin olin alun perin asettanut tavoitteekseni.

Tulevaisuudessa olisi mielenkiintoista tutkia käytännössä, kuinka DAD prosessiviitekehys vaikuttaa organisaatioiden projektien lopputulokseen vertailemalla lopputulosta aiemmin joko puhtaasti ketteriä toimintatapoja käyttäneissä organisaatioissa tai perinteistä ja ketterää tapaa yhdistäneissä organisaatioissa. Ovatko ratkaisut parantaneet asiakastytyväisyyttä, nouseeko organisaatioiden saama arvo toimitetun ratkaisun myötä, kehittyvätkö organisaation toimintatavat läpi organisaation. Toisaalta olisi mielenkiintoista tutkia syvällisemmin, minkälaista osaamista tulevaisuudessa tarvitaan ja minkälaisista muutosta tarvitaan nykyisiin organisaatioihin, jotta vuosikymmenen sisällä tietotekniikkaan liittyvät projektit onnistuvat jouhevammin ja tuottavat enemmän todellista hyötyä organisaatioille. Edelleen jatkossa voisi kehittää organisaatioille käyttöönottoppaan, kuinka hyödyntää DAD viitekehystä ensiaskeleesta alkaen.

Lähteet

Ambler, W.S. 2014. Full Agile Delivery Lifecycles Luettavissa:

<http://disciplinedagiledelivery.wordpress.com/lifecycle/>. Luettu: 7.5.2014.

Ambler, W.S. 2010. Disciplined Agile Delivery (DAD) Lifecycles. Luettavissa:

https://www.ibm.com/developerworks/community/blogs/ambler/entry/disciplined_agile_delivery_dad_lifecycle14?lang=en. Luettu: 5.3.2014.

Ambler, W.S. 2012. Roles in Disciplined Agile Delivery. Luettavissa:

<http://disciplinedagiledelivery.wordpress.com/category/dad-roles/>. Luettu: 6.5.2014.

Ambler, W.S. & Lines, M. 2012 Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise. International Business Machines Press. Crawfordsville.

Auer, A., Auer, L., Heinäsmäki, M., Hölttä, J., Kalliala, E., Laanti, M., Laine K., Lekman, L., Miinalainen, P., Naski, H., Piiparinen, T., Puhakka, H., Pyhäjärvi, M., Pääkkönen, T., Räisänen, S., Sora, H., Taipale, M., Talvio, J., Tanninen, A., Toikkanen, T., Toivola, T., Toro, K., Valsta, A., Väyrynen, V. & von Weissenberg, M. 2013. Ketterää kehitystä. Finn Lectura. Helsinki.

Begel, A., Bird, C., Murphy, B., Nagappan, N., Williams, L. & Zimmerman, T. 2013.

Have Agile Techniques been the Silver Bullet for Software Development at Microsoft?

Luettavissa: <http://www.cabird.com/papers/murphy2013agile.pdf>. Luettu: 24.4.2014.

Clarce, P., O'Connor, R.V. & Ylimaz, M. 2012. A systematic approach to the comparison of roles in the software development processes. University of Limerick. Luettavissa: http://doras.dcu.ie/17072/1/10.1007%3A978-3-642-30439-2_18.pdf. Luettu:

6.2.2014.

Forsberg, K., Mooz, H. & Cotterman, H. 2000. Projektinhallinta. Edita Publishing Oy. Helsinki.

Haarala, T. & Mäkilouko, M. 2013. Projektitoimintamallin kehitys PK -yrityksessä. Projektitoiminta, 2, s. 86.

Hartnett, G. 2011. Agile for any organization – a guide. Luettavissa: <http://www.goshido.com/category/blog/guides/>. Luettu: 21.4.2014.

ICT Standard Forum 2012. Projektien johtaminen - Johdanto. Luettavissa: <https://www.tietohallintomalli.fi/malli/projektien-johtaminen/johdanto>. Luettu: 5.2.2014.

Karlström, D. & Runeson, P. 2006. Integrating agile software development into stage-gate managed product development. Luettavissa: <http://alarcos.inf-cr.uclm.es/doc/MetoTecInfInf/casosestudio/case%20study%201.pdf>. Luettu: 10.2.2014.

Kuikka, M. 2008. Ohjelmistoprojekti aloittavassa ohjelmistoyrityksessä. Pro-gradu - tutkielma. Itä-Suomen yliopisto. Luettavissa: ftp://cs.joensuu.fi/pub/Theses/2008_MSc_Kuikka_Mika.pdf. Luettu: 5.2.2014.

Laanti, M. 2013. Miksi toinen projekti sujuu ja toinen takkuu? Projektitoiminta, 2, s. 50.

Lehtonen, P. 2013. Määrittelyn haasteet siirryttäessä vesiputousmallista ketteriin menetelmiin. Opinnäytetyö. Haaga-Helia ammattikorkeakoulu. Helsinki. Luettavissa: http://www.theseus.fi/bitstream/handle/10024/61825/Lehtonen_Piia.pdf?sequence=1. Luettu: 6.2.2014.

Lekman, L. 2013. Ketterästi johtamalla enemmän onnistumisia. Projektitoiminta, 2, s. 40.

Marchewka, J.T. 2013. Information Technology Project Management. Wiley. Singapore.

Markkula, M., Mustajärvi, O. & Suurla, R 2002. Developing and Implementing Knowledge Management in the Parliament of Finland. Luettavissa: http://www.eduskunta.fi/fakta/vk/tuv/KM_Finnish_Parliament.pdf. Luettu: 27.2.2014.

Partiowiki 2013. Robert Baden-Powell. Luettavissa: http://fi.scoutwiki.org/Robert_Baden-Powell. Luettu: 5.3.2014.

Penttonen, P. 2013. Projektinhallinta ketterässä sovelluskehityksessä. Pro-gradu -tutkielma. Itä-Suomen yliopisto. Kuopio. Luettavissa: http://epublications.uef.fi/pub/urn_nbn_fi_uef-20130537/urn_nbn_fi_uef-20130537.pdf. Luettu: 3.2.2014.

Roukala, S. 2011. Scrumin käyttö sovelluskehitysprojektissa. Amk-opinnäytetyö. Laurea-ammattikorkeakoulu. Leppävaara. Luettavissa: http://www.theseus.fi/bitstream/handle/10024/32272/Roukala_Selja.pdf?sequence=1. Luettu: 24.1.2014.

Ruuska, T. 2012. Vaatimusmäärittely ketterässä ohjelmistokehityksessä. Pro-gradu -tutkielma. Jyväskylän yliopisto. Jyväskylä. Luettavissa: <https://jyx.jyu.fi/dspace/bitstream/handle/123456789/38590/URN:NBN:fi:ju-201209202463.pdf?sequence=1>. Luettu: 3.2.2014.

Sethi, R. 2013. How to Handle Complex Non-Functional Requirements in Agile Projects. Luettavissa: <http://blog.sdelements.com/how-to-handle-complex-non-functional-requirements-in-agile-projects/>. Luettu: 3.2.2014.

Solanterä, T. 2010. Projektinhallintamenetelmien pm methodology ja scrum yhdistäminen. Luettavissa: http://www.theseus.fi/bitstream/handle/10024/12459/Solantera_Terhi.pdf?sequence=1. Luettu: 15.2.2014.

Suomen Projekti-Instituutti 2013. Projektijohtamisen sanastoa. Luettavissa:
[http://www.projekti-instituutti.fi/osaamisen_kehittaminen/projektijohtamisen_](http://www.projekti-instituutti.fi/osaamisen_kehittaminen/projektijohtamisen_sanastoa)
sanastoa. Luettu: 6.2.2014.

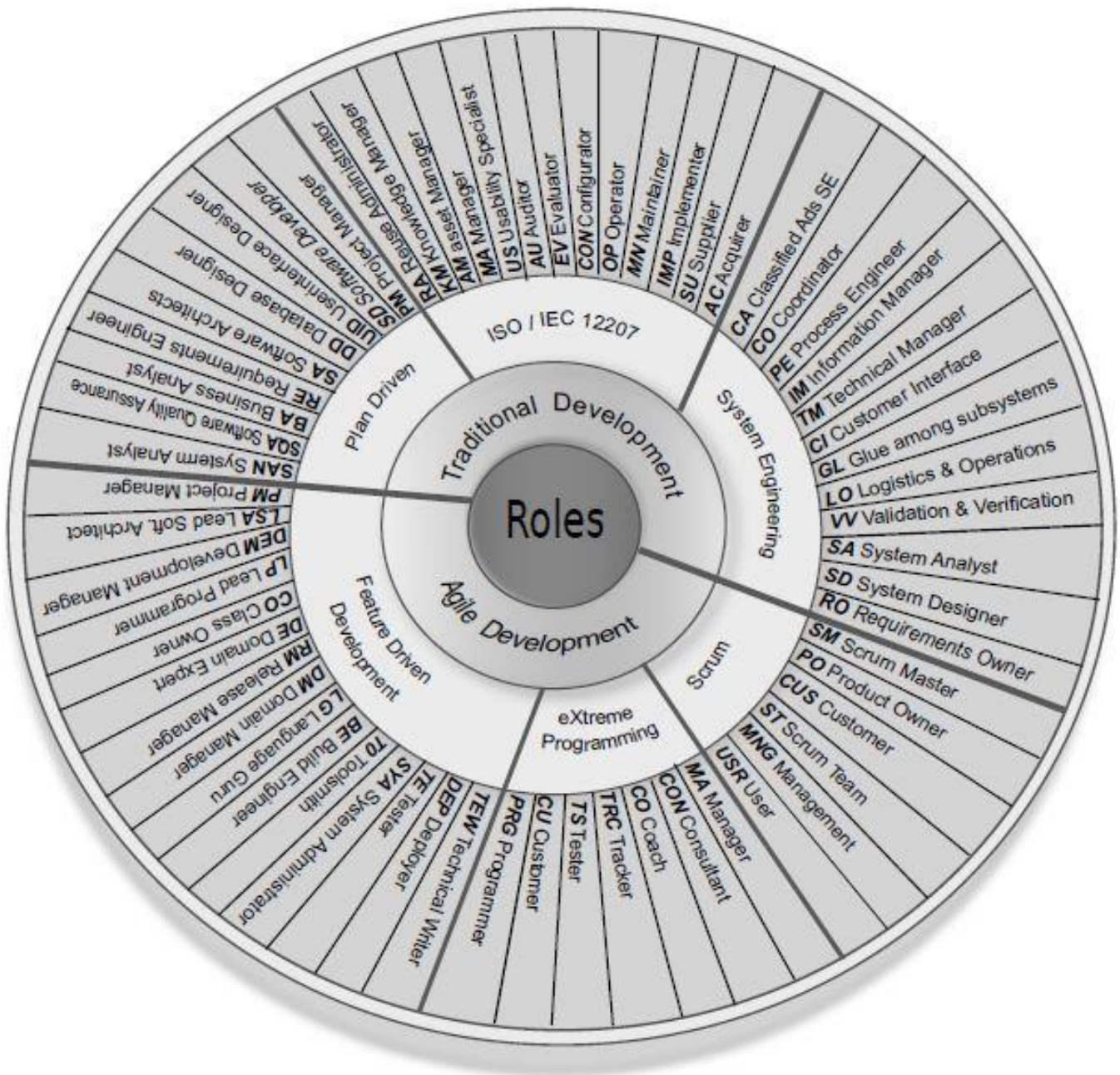
Szalvay, L. & James, M. 2013. Agile Methodology. Luettavissa:
<http://agilemethodology.org/>. Luettu: 25.1.2014.

The Standish Group 2013. Chaos Manifesto 2013 Think Big, Act Small. Luettavissa:
<http://versionone.com/assets/img/files/ChaosManifesto2013.pdf>. Luettu: 19.2.2014.

Tonnquist, B. 2012. Project Management. Sanoma Utbildning AB. Tukholma.

Liitteet

Liite 1. Perinteisen ja ketterän ohjelmistokehityksen rooleja (Clarce ja muut 2012, 11)



Liite 2. Termit ja selitteet

agilisti	ketterien käytänteiden osaaja
Agile Manifesto	Ketteryyden julistus
fixed price	alkuarvioon perustuva rahoitusmalli
DAD	Disciplined Agile Delivery
osaprojekti	summaprojektin osa, esim. sovelluskehitysosuus
projektin elinkaarimalli	kuvaus tapahtumien järjestyksestä ja tuotoksista
summaprojekti	kokonaisuus, jossa yhdistyvät kaikki osaprojektit
stage gate	vaiheperusteinen rahoitusmalli
time and materials	jatkuvan rahoituksen malli
Work items list	DAD tuotekehitysjono